Supporting Information

Kassal *et al.* 10.1073/pnas.0808245105

SI Text

Quantum Resource Estimation. There are 2 kinds of quantum register in the simulation algorithm: the state registers, used to store the wave function, and the ancilla registers, used to store the potential energy and the intermediate calculation results. If we assume a simulation of a d-dimensional system in which each Cartesian coordinate is divided into a uniform grid of $N = 2^n$ points, the representation of the wave function requires a total of nd qubits in d registers. As for the ancilla registers, their total number will depend on the complexity of evaluating the potential and the kinetic energy. At least 1 register is always required, to be used as the target of addition for the purpose of phase kickback. The ancilla registers will require m qubits each, where m is chosen in such a way that the registers can store the value of $V(x)$ with desired accuracy, in the form of a binary integer between 0 and 2^m.

The time required for the simulation is the number of elementary (1- and 2-qubit) gates required to perform the algorithm. Except in trivial cases, the evaluation of the potential energy will be much more complicated than that of the kinetic energy T, which is a simple quadratic form. We therefore approximate the total gate count as being equal to the complexity of evaluating the potential: Even for the simple Coulomb potential, the error thus introduced to the resource count is substantially $<1\%$.

Coulomb Potential. The simulation of chemical dynamics depends on computing the Coulomb potential, and here, we provide a detailed count of the resources required for evaluating it on a quantum computer. We begin by developing some necessary quantum arithmetic.

For addition, we adopt Draper's quantum addition algorithm (1), which is based on the quantum Fourier transform (QFT), and requires only $\frac{3}{2}m^2$ controlled rotations. Although it is not asymptotically optimal (i.e., it scales as $O(m^2)$ and not $O(m)$ as does the schoolbook addition algorithm), it both has a small prefactor that makes it attractive for the addition of small numbers, and it is easily adapted for multiplication. Subtraction requires the same number of rotations, except that they are performed in the opposite direction.

We perform multiplication using the schoolbook method. The first multiplicand is repeatedly bit-shifted and added to the accumulator if the corresponding bit of the second multiplicand is 1. Because each number has m bits, we need to make a total of m such controlled additions (C-ADD). The product will have $2m$ bits, but we will keep only the m most significant ones, essentially performing floating-point arithmetic. For the C-ADD, we first apply a QFT to the accumulator, as in Draper's algorithm (we will also apply an inverse QFT at the end, and these 2 require n^2 steps in all). Each C-ADD requires $\frac{1}{2}m^2$ CC-rotations, which each can be implemented by using 2 CNOTs and 3 C-rotations. Hence, each C-ADD requires $\frac{5}{2}m^2$ operations, giving a total of $(\frac{5}{2}m^3 + m^2)$ gates for a multiplication. However, because half of the CC-rotations are to the insignificant bits of the accumulator and are subsequently discarded, we need to perform only $(\frac{5}{4}m^3 + m^2)$ gates for a multiplication.

To compute the Coulomb potential, the distance r between 2 particles, $r^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2$, must be known. Evaluating r^2 requires 3 subtractions and 3 squarings (the 2 additions are performed automatically because the squarings are really additions that can use the same accumulator). For squaring, we multiply the number by itself using the multiplication circuit, giving the total requirement of $(\frac{15}{4}m^3 + \frac{15}{2}m^2)$ gates for computing r^2. The same computation would be used in momentum space for computing p^2 or for simulating a harmonic oscillator potential.

The evaluation of the Coulomb potential is complicated by the need for a square root. Because evaluating $\sqrt{S}$ is just as difficult as evaluating $1/\sqrt{S}$, we can find $1/r$ from r^2 in 1 computation using the Newton–Raphson method, with the iteration $x_{n+1} = x_n(3 - r^2x_n^2)/2$. The number of iterations will depend on the desired final accuracy, but numerical experiments show that for many ranges of S, 4 iterations suffice to compute $1/\sqrt{S}$ to within $<0.03\%$ over the entire range. Each iteration requires 1 subtraction and 3 multiplications (one of them bit-shifted due to the factor of $\frac{1}{2}$). So the requirement for $1/\sqrt{S}$ is $(15m^3 + 18m^2)$ gates, which, together with calculating the distance r^2, gives the total requirement for the Coulomb potential as $(\frac{75}{4}m^3 + \frac{51}{2}m^2)$ gates for each pair of particles.

Potentials Fitted from First-Principles Calculations. When using the Born–Oppenheimer approximation, one uses a potential $V(\mathbf{x})$ that is a function of only the nuclear coordinates. It is the total energy of the molecule assuming that the electrons are in their ground state given the potential induced by the nuclei at coordinates $\mathbf{x}$. In general, this ground-state energy is difficult to compute on a classical computer. Thus, interpolation schemes may be used to approximate $V(\mathbf{x})$. Here, we analyze the computational resources needed for such schemes.

We represent the potential as a d-dimensional interpolating polynomial:

$$V(\mathbf{x}) = \sum_{k_1,k_2\ldots k_d=0}^{K} c_{k_1k_2\ldots k_d} x_1^{k_1} x_2^{k_2} \cdots x_d^{k_d}$$

$$= \sum_{k_1=0}^{K} b_{k_1} x_1^{k_1} \sum_{k_2=0}^{K} b_{k_1k_2} x_2^{k_2} \cdots \sum_{k_d=0}^{K} {}_{k_1k_2\ldots k_d} x_d^{k_d},$$

which can be evaluated using Horner's method starting with the innermost sum. That is, one has to evaluate 1 order-K polynomial in x_1, K such polynomials in x_2, and so on until K^{d-1} polynomials in x_d. That is, the total number of polynomials that need to be evaluated is $\sum_{i=1}^{d-1} K^i \approx K^d/(K-1)$. In order that the results of these calculations will be available as constants for higher-level polynomials, all of the intermediate polynomial evaluations have to be saved in temporary memory along the way. The number of required registers is $K^{d-1}/(K-1)$.

Each polynomial that is evaluated has the form

$$P(x) = \sum_{k=0}^{K} p_k x^k = p_0 + x(p_1 \cdots + x(p_{k-2} + x(p_{k-1} + xp_k))),$$

where p_k is some constant. Therefore, each polynomial evaluation requires precisely K additions and K multiplications. As we have seen above, an addition requires $\frac{3}{2}m^2$ operations and a multiplication $(\frac{5}{4}m^3 + m^2)$, meaning that, in total, each polynomial evaluation requires K $(\frac{5}{4}m^3 + \frac{5}{2}m^2)$ gates. Therefore, the total number of gates required to calculate V is K^{d+1} $(\frac{5}{4}m^3 + \frac{5}{2}m^2)/(K-1) \approx K^d$ $(\frac{5}{4}m^3 + \frac{5}{2}m^2)$. Furthermore, the total qubit requirement is $n_{\text{tot}} = nd + mK^{d-1}/(K-1)$.

1. Draper TG (2000) Addition on a quantum computer. *arXiv*:quant-ph/0008033.